DOCUMENT RESUME

ED 331 562 JC 910 243

AUTHOR McCready, Joan W.
TITLE Computer Literacy: How Much Do You Really Need To Know?
PUB DATE 17 Nov 90
NOTE 34p.; Edited transcript of a paper presented at the Annual Convention of the Missouri Association of Community and Junior Colleges (25th, Lake Ozarks, MO, November 15-17, 1990).
PUB TYPE Guides - Non-Classroom Use (055) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS College Planning; Community Colleges; *Computer Assisted Instruction; *Computer Literacy; Computer Software; *Computer Uses in Education; Education Work Relationship; *Microcomputers; Program Implementation; Two Year Colleges; Vocational Education

ABSTRACT

Designed for community college faculty who plan to employ computers in their classrooms, this paper defines computer literacy as not only knowing how to use computers but also knowing when it is appropriate to do so. The paper provides examples of inappropriate uses of computer technology in education, presents a brief history of the microcomputer, offers an overview of programming and software languages, and explores the implications of computer technology for community colleges' diverse student body. Next, the following suggestions concerning computer use in the classroom are discussed: (1) know what the computer requirements of four-year colleges and/or industry are for graduates; (2) encourage the use of word processing software in writing assignments; (3) be realistic about student access to computer equipment; (4) assess a program's ability to enhance teaching quality or students' understanding sufficiently to justify its use; and (5) centralize campus purchasing and equipment usage decisions. The paper concludes by suggesting that community colleges should develop computer information-sharing networks with four-year faculty, encourage career and vocational faculty to stay abreast of changes in computer technology, adapt continuing education computer classes to meet students' varied needs, carefully design and manage college computer labs, and create college-wide computer committees to oversee the use of computer technology. Appendixes include a list of software evaluation sources, a guide to word processors, a software acquisition checklist, a discussion of computer needs by program, and additional classroom tips. (JMC)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED331562

Computer Literacy: How Much Do You Really Need to Know?

Joan W. McCready

Paper presented at the Missouri Association of Community and Junior Colleges Annual Convention "Celebrating our Uniqueness" (25th, Lake Ozarks, Missouri, November 15-17, 1990).

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

J. W. McCready

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☒ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

JC 910243

"True computer literacy is not just knowing how to make use of computers and computational ideas. It is knowing when it is appropriate to do so."

Seymour Papert

Computer Literacy: How Much Do You Really Need to Know?

Joan McCready, presenter

Computer Programmer I, Computer Center; Part-time Instructor, Computer Information Systems

Jefferson College
PO Box 1000
Hillsboro, Missouri 63050

Check List for Computer Use

* Know what the computer requirements of four-year colleges and/or industry are for your graduates.
* Encourage the use of word processing software in producing written projects.
* Be realistic about student access to computer equipment.
* Does any one program enhance your teaching or your students' understanding sufficiently to justify using it?
* Centralize campus purchase and equipment usage decisions to allow for maximum equipment use and minimum expenditures.

MACJC 25th Anniversary Celebration: Celebrating Our Uniqueness

Breakout Session #30, Saturday, November 17, 1990, 8:30 am

from *The Chronicle of Higher Education*

FORWARD

The text of this paper is an edited transcription from the tape of my presentation at the Missouri Association of Community and Junior Colleges (now the Missouri Community College Association), given last November.

I spoke that day from an outline, not a prepared paper, so this transcription uses spoken idioms more than "scholarly" grammar. Incomplete sentences have been completed, while redundancies and extraneous comments have been removed.

While all of the text in the handout is included in the paper, it is not in the form of the handout given to the attenders. Where appropriate, material presented in overhead projections has been incorporated directly into the text at the point where the overhead was used; otherwise, handout materials appear in the appendices.

Joan McCready
May 31, 1991

This is a live recording from the twenty-fifth anniversary convention of the Missouri Association of Community and Junior Colleges, entitled *Celebrating our uniqueness: A banner year*. Held November 15 through the 17, 1990 at Lodge of the Four Seasons in Lake of the Ozarks, Missouri.

I want to thank all of you for coming to this session this morning. My name is Joan McCready. I work at Jefferson College as a computer programmer in our computer center. I am also a part-time instructor in our computer information systems program. I primarily teach Introduction to Data Processing, which is our introductory-level, computer literacy-type class. I have also taught B.A.S.I.C., Lotus, and Microcomputer Software[1] one semester each. I've been teaching the Introduction class for about four and a half years.

One of the reasons I proposed this topic to the MACJC programming committee came about last spring. I was serving on one of the long-range planning subcommittees at our college. We kept talking about computer literacy in the classroom, and what needs we have. People started talking hardware and software; everybody should do this; everybody should have a Macintosh; everybody should be teaching WordPerfect, even if they're in the Math department. I realized there is a lot of confusion about what you really need to know to use computers in the classroom.

I do want to apologize to any administrators or classified staff[2] who are here. My talk is primarily targeted this morning toward faculty. I think you can get something out of it, but I geared the talk toward the classroom use of computers. I've been asked to do something similar for the classified staff workshop at Jefferson College in the spring which will be more appropriate for the classified staff.

When the call came out for MACJC I found myself sitting at the word processor typing a proposal, and I thought, "do I really want to do this?" and before I could say no, I mailed it off. So that is why I am here this morning.

True computer literacy is not just knowing how to make use of computers and computational ideas. It is knowing when it is appropriate to do so.

Let me start off with a quote about what computer literacy is. There is a lot of controversy and a lot of discrepancy, a lot of in-fighting about the definition of computer literacy. I know there are several computer faculty here; I was hoping none would show up because I may be speaking some heresies. In any event, I came across this quote this summer and I really liked it. "To be truly computer literate you have to not only know how to use the technology, but when to use the technology appropriately."[3] When I was going over this talk, my husband said, "Well, give some examples of inappropriate uses."

One obvious inappropriate use of the computer is to buy a computer to balance your checkbook. If you don't balance your checkbook with a pencil and pocket calculator, you're not going to balance your checkbook on the computer; you're not going to spend the time and effort and be organized enough to do the data entry that is needed.

A classroom parallel to this would be the person who wants a computer on his desk to figure grades and to keep a grade book, but he/she is a graduate research faculty member at a major university, teaches one class a year with six students. They're all graduate research students, so there is one major paper and no tests, but the professor has to have a computer to figure his grades. That is an inappropriate use of computers.

Educational software (elementary and secondary) has a long and checkered history, most of it bad. Many of these programs are little better than electronic flash cards. For a third or fourth grader who is hyperactive, you **might** be able to keep his/her attention long

enough to learn arithmetic, provided he is allowed to shoot down something when he gets a right answer. You might be able to keep his attention and use a computer as an electronic flash card. But at the college level, our students should be beyond that. Not that there isn't a place for a drill- and-practice program, but it needs to be more than that if you're going to use computers in the classroom setting.

A different example of inappropriate academic computer usage is, you might have an excellent program - a very excellent program - a piece of software that makes a point, that teaches the student the concept. But it takes three and a half hours to get the student to understand how to run the program and to get the data entered for the program; takes only ten minutes to actually run the program. And that is out of the entire semester's worth of work. That is an inappropriate use of technology; not that it isn't a good program, but the effort needed to use it outweighs the benefit that it brings to your students. So that is some examples - you may have some other examples - of inappropriate uses of the technology.

Another problem of computer literacy is, people who don't understand computers think that the computer itself does things, that the computer itself has a mind of its own. We muddy the waters in computer courses because we talk about memory; we talk about the computer processing data into information, and many times the students do walk away thinking the computer can do things on its own. It really can't. The computer is a tool, just like a slide rule is a tool, a typewriter is a tool, tape recorders in a language lab are tools, that we use to help and teach, but we really are still in charge of the data, not the computer despite what this...

Illustration A

I love this cartoon (illustration A), which came from the *Chronicle of Higher Education*, and I am sorry I didn't get the date, sometime over the summer. So we tend to think that computers have a mind of their own, but they really don't. We are the mind behind the computer. Computers merely help our minds work faster.

Let me see a show of hands; this is going to be biased because I know there are a lot of computer-literate people sitting in the audience. How many people think that the IBM PC was introduced in the mid-sixties?

How many think it was introduced in the mid-seventies?

Would you believe 1981?

The IBM personal computer revolution is only ten years old; not quite ten. I thought this might give us a little historical perspective (appendix 1). This isn't in your handout; I apologize. I came up with this Wednesday morning while I was waiting to do something else.

Dr. Kemeny, of Dartmouth, developed B.A.S.I.C. in 1965; it was still running on mainframe computers as an educational teaching tool. It was the first programming language that was really intended for the general student, the non-computer specialist, to learn something about computers. It still is the most widely used computer programming language, primarily because when the PCs first came out, the companies had the good sense to bundle B.A.S.I.C. right with the machines as the language to use. The invention of the

microprocessor in '71, or maybe '69, by Dr. Ted Hoff, or maybe somebody else (there is a dispute here, in fact a lawsuit was just settled a couple of weeks ago on that)... The microprocessor drives - is - the heart of the personal computer; therefore, when the microprocessor was invented the personal computer became possible. You were no longer necessarily tied to a mini- or a mainframe computer.

Nineteen-seventy-two, the first commercial kit for a microcomputer. It probably did less than our current hand-held calculators that you get at WalMart for $5.99. In 1975, Steve Jobs and Steve Wozniak built their first computer, the Apple I[4]. Nineteen-seventy-seven was a big year for personal computers -- Radio Shack came out with its TRS-80; Commodore with its PET; Atari, which was making a killing in the video-game market and the pinball market, came out with its first computer; and the Apple II series made it out of the Jobs' garage and into the marketplace.

In 1979, a very important event for the business world occurred. Two Harvard M.B.A. students decided that they were tired of using manual spreadsheets, which included erasing and recalculating, in accounting. Accounting had been using manual spreadsheets for many, many years, probably centuries, but you had to erase and recalculate everything by hand. Since these two had an Apple II computer, they wrote VisiCalc for the Apple. VisiCalc, the first electronic spreadsheet program, suddenly made microcomputers of worth to the business community.

MS-DOS in 1980; in August 1981 the IBM personal computer first reached the marketplace. Nineteen-eighty-three - Lotus 1-2-3, which has replaced VisiCalc (VisiCalc is no longer in business, to the best of my knowledge), was introduced. In 1984 the Macintosh

first appeared, the first major challenge to the IBM PC. (When IBM introduced their personal computer, they set the industry standard for personal computers.)

Today most of these other companies have fallen by the wayside. Radio Shack still sells a TRS-80, but they also have IBM PC clones. Commodore still has their Commodore 64 and their 128, and they are still doing well as educational and home machines, but they also introduced several years ago an IBM PC clone. Atari got out of the computer business and returned to doing video-games, which they do real well.[5] There many other microcomputer systems that have fallen by the wayside or have jumped on the IBM PC bandwagon.

In 1987, the 80386 microprocessor first reached the marketplace; it is now the business standard. In 1988 IBM introduced the P/S2 series, using O/S2, which fell flat on its face when it was not MS-DOS compatible. Those machines have now been re-tooled to also be MS-DOS compatible, but the O/S2 operating system is in big trouble.[6] So that is just a brief overview of where we are today in the use of computers.

There have been many predictions about what would happen in the business world with computers. When the International Business Machine Company first got interested in buying some of the computer technology and marketing it, buying it from the university researchers and marketing it to the business world, Thomas Watson, the president and founder of IBM, said, "I think there is a world market for about five computers."[7] My husband sees nothing wrong with that statement. He still thinks that the world only needs about five computers.

(I am going to make many references to my husband, since he gave me most of my anecdotal material for this talk today.)

Contrast that with the statement that was made by an educator several years ago. She said, "I am anxiously awaiting the day when these students are college freshman, and they walk into their computer science course with ten year's of programming already under their belts at age 18. The implications for their futures, and ours, staggers the imagination."[8] She was talking about computer majors, but these same students are going to walk in to our English classes, our history classes, our data processing classes, our math classes, and they're going to be computer literate. At the community college level we are going to have a much wider span for a much longer time. We are going to get the 18-year-olds who have been programming for ten years, who have owned in their short life maybe three computers, one of which they built themselves. But we are also going to get the 50-year-old worker who has been laid off from his job, his union has said "Hey, buddy, you had better retrain; you're not going to come back to work in the auto industry" or some other manufacturing industry. We're going to get the housewife who dropped out of high school to raise a family and now finds herself divorced against her will and having to support her family by herself and having to retrain for today's marketplace. So we are going to have a much wider span of students that we need to help into the computer age along the way. We are going to have ourselves, the faculty members who by-and-large had little exposure to computers in their undergraduate or graduate education, depending on how long ago they went through the system.

Know what the computer requirements of four-year colleges and/or industry are for your graduates.

So where do we start in our checklist for computer use in the classroom? I think the first thing that we in the community college have to keep in mind is that we are vocationally oriented. Even in the liberal arts, even in the arts and sciences and the transfer programs, we need to get our students "hired" by a four-year curriculum so that they can graduate in a timely manner from their baccalaureate program. We need to keep an eye at all times on what the baccalaureate programs and our transfer programs are doing. We cannot be innovative -- and that is a shame in a sense, because I think we have more potential for being innovative at the community college because we work in a local setting.

Our career areas, our vocational programs can be more innovative because we are not gearing our students to go on to further education; we are gearing our students to go out into the workplace. That is where we can meet the needs of our community best. We can gear students to that special industry that is fifty miles from us, and we don't have to worry about curriculum requirements for further education. But we still have to keep an eye on what industry needs in the way of technology, so in both the transfer and vocational sides, we need to keep an eye on where our students are going from here.

If the four-year schools were honest, they would be doing this also. Everybody is vocationally oriented, whether our students are going to get a job when they graduate from us or when they get their PhD fifteen years from now. They still want to get a job.

We need, first of all, to know what the requirements are. I gave you, in your handout, a list of some software acquisition groups (appendix 2). These groups do sell some

specialized software, but for the most part are academic clearinghouses for software educators, like yourselves, who have written the software to meet a need in their classrooms, who really aren't concerned with selling it in the marketplace, so these clearinghouses provide the service of distribution. The higher education community lost a very important clearinghouse for Macintosh software when Kinkos decided, after too many lawsuits for copyright violation, that they would no longer carry professors' software. That was mostly on the Macintosh, and I believe they discontinued that service last spring sometime.

Question: Who was violating the copyright?
Answer: Apparently, some of the faculty members made their packages look too much like some commercial packages. I don't think there was deliberate intent, but if you're trying to teach a class in Lotus 1-2-3, for example, you do want your package to look like Lotus 1-2-3, and the Lotus Corporation thought, well then, you should be using Lotus 1-2-3. Those have been some of the problems there.

The list of software acquisition groups came from a *Chronicle of Higher Education* article and it just gives you some information if you would want to pursue software acquisition, particularly arts and science faculty. Many times, faculty in our transfer programs don't know where to go to look for software. There is some technical software listed in here; in particular, NASA makes a lot of its software that it has developed available to science faculty. We first of all have to keep an eye on what is happening in the real world, be it our transfer four-year colleges or industry.

One way you can encourage computer literacy, and I think this probably the most important reason to become computer literate, is to encourage the use of word processing software from your students, and to learn to use a word processor yourself. In fact, I had

Encourage the use of word processing software in producing written projects.

originally written that you **insist** upon the use of word processing software, but there are some problems with insisting upon it. You need to encourage the use of word processing software because in academe our job is to turn out papers. I sometimes think that turning out papers takes precedence over teaching classes or getting anything else done. You have to turn out a memo or position paper or something like that.

Joke: How can you tell when the faculty have been using the computers for word processing?
Answer: There is White-Out™ on the screen.

In academe, overall, the publish-or-perish mentality is there. The ideas is not only to do research, to add to the body of knowledge, but also to disseminate that information to the rest of your colleagues. At the community college level we don't do a lot of publish-or-perish, but you are **encouraged** to publish, and you do turn out a lot of typewritten pages throughout a semester, regardless of whether you are in a very technical field or whether you are in the most traditional liberal arts fields. You are still going to be writing papers; if nothing else, turning out tests. The students going into business and industry, going out into the job market, are going to be expected to know how to write papers. In English composition, writing across the curriculum is very big at the moment, and more and more the business world is expecting communications to be in typewritten, neat, word-processed form. You should, then, encourage the use of word processors.

I hated high school. Most people say that was the best years of their lives. But there was one class I took in high school that has earned me more money than any other class I have taken since. That was Typing I. The need to learn to use a typewriter is being replaced today by the need to learn a word processor. (You do have the problem of lab

access, which I will talk about in a little while.) Let's take a look at what these word processors should be (appendix 3).

Your general student -- your English major, your history major, your music major, your math major, your physics major, your pre-engineering major -- should have access to and learn to use a simple word processor. That simple word processor should allow you to move copy, delete, and insert text. It should include a spelling check routine; then you must teach the student that just because they ran a document through the spelling checker does not mean it is correct. The basic word processor should be able to print underline, italic, and boldface types, and be able to change characters per inch in the middle of a document. That is your basic, simple word processor. With that, students can write papers up through and including their doctoral dissertation. My husband did his doctoral dissertation using Appleworks. Including, believe it or not, crunching his statistics on the Appleworks spreadsheet. That was quite a trick to massage all his data given the memory limitations of Appleworks and the Apple IIe. That is all your basic student really needs; that is all you really need in a word processor.

Your office systems technology students, your secretarial students, and your data processing/microcomputer software specialist students need access to a full-featured word processor. This program would include footnote and endnote placement; automatic index creation; basic graphics capabilities -- those are your boxes, lines, and shadings similar to the handout I gave you today; and they should have access to a laser printer for final document printing. That doesn't mean every microcomputer needs a laser printer attached to it, it just means there needs to be one that is available for the final copy. Your office systems

technology and word processing specialists also need to be trained on desktop publishing software and equipment. Any graphics artist or fine arts students would also benefit from learning desktop publishing.

This is a guide to what basic student needs are. If you can put WordPerfect on every machine, if you can take the time to train every student to use all the features of WordPerfect, go to it. We don't have that much lab time, quite frankly, and we are, in fact, still using the Apples quite a bit, particularly for our arts and science students, because Appleworks is a basic, simple word processor.

Be realistic about student access to computer equipment

That brings us to student access to the computer equipment. Access problems is the reason why I changed "insist on using word processing" to "encourage the use of word processing." Every school is facing the problem with access to computer equipment, whether it is Three Rivers[9] with their one computer lab, or whether it is Jefferson College with - I think we have five or six IBM labs (and they were hoping to install another lab this weekend), two Apple labs, a CAD-CAM lab, and I have lost track of everything else we do have. We still have problems getting access to the computers. This is why we still use the Apples at Jefferson College. We can't afford to give up those machines, and we can't afford to give up that lab time. Some of the four-year schools require the students to buy their own computer. It is part of their tuition for their freshman year or it is factored in over the four years; if they leave the school before that time the school repossesses the computer. That would be the ideal. In reality, we know

that our students sometimes have trouble scraping together an extra $60 for a workbook to go along with the textbooks that their grants cover.

Current demand for computer classes will dictate when the labs are available to students. If the students do not have access to a computer at home or in the workplace, you have to make open lab time available to them outside of class. Right at the moment, we can't do that very often. There are very limited times during the week when the computer labs are available, when there is not a class scheduled in there. We are even using our computers on Sunday afternoon. If the adult education division wants to offer a five-week Lotus overview class, they have to do it on Sunday afternoon. It is the only time available.

Question: What is the rate of breakdown?
Answer: We try to keep it minimal, and I will address that in a later point.

Question: Is lab included in the class?
Answer: Again, that depends upon the class. Our Introduction to Data Processing class does have some lab time factored in, but most of their work is done outside of class. Our Microcomputer Software Applications class is a lab class; four hours in the lab for three hours credit. Any lecturing is done in the lab as part of the lab exercise. It varies anywhere between those two extremes. On the other hand, if a student wants to use the word processor for an English composition class, that is fine and dandy, but there is absolutely no lab component whatsoever in that class. In-class lab time varies from both extremes depending on the particular class.

Summing up to this point, then, we've talked about knowing what the computer requirements of industry are, and being realistic about the student access to the equipment.

When you acquire software for classroom use (appendix 4), first of all learn which software packages are standard in your field, which ones industry wants your students to know when they walk in and apply for a job. I was really appalled at this attitude when my husband was doing doctoral studies at Indiana University. I was, of course, looking for work on campus; when I applied for several jobs, they said, "Do you know WordStar?" I

said, "Well, I've used a word processor before," and they replied, "No, do you know WordStar? If you don't, don't even bother to apply." They wanted somebody who already knew the package, not somebody who was capable of learning the package and had some background and experience. That may be an extreme, but this is what your students are going to face when they walk in a door and file an application, even for a summer job.

Does any one program enhance your teaching or your students' understanding sufficiently to justify using it?

Next, you want to ask yourself, would using this software enhance or distract from teaching your class. If you are a computer science instructor, the software is the content of your class. If you are teaching word processing, then WordPerfect 5.1, or WordStar or whatever you are using, is the content of the class; it is the textbook. But if you are teaching English Composition and you decide to have a five-week presentation on how to use WordStar and how to use all the full-features of desktop publishing, that is a distraction from the content of your class, which is to teach English Composition. In the math department, if you are going to spend six weeks out of a 16-week semester teaching the students to write a B.A.S.I.C. program to solve one problem that you could solve on the board in 15 minutes, that is a distraction from teaching your class. It may be an educational experience, but it is not the one that your students signed up to learn from that particular class.

With access to lab time being a problem, does the learning experience provided by the software outweigh the time needed to learn to use the software effectively. Even if a particular program is the standard software package in your field, what machine does it run

on? Does your school have any of those machines? Do you have the faster processor to run some of the graphics intensive packages coming out today? We had a problem where one of our physics teachers found a package that would do just marvelous things for his physics students. Unfortunately, it ran only on the Macintosh. We have two Macintoshs on campus, both in Administrative offices. If someone wants to donate $20,000 for us to add a Macintosh lab, I think the physics instructor would be eternally grateful to you. Until that happens, he is going to continue to teach that particular concept from the book and on the board.

Even if you do have the right machines on campus, what is the competition for that lab time? How many other instructors would also need access to this type of computer during your class time? Would the students have adequate access and knowledgeable support to use this package outside of class? It is not just enough to tell the students to go over to the computer lab in their spare time and work through this program; there has to be a lab assistant on duty who knows the program and can get them out of difficulties, and can understand whether it is working correctly or not. It would also be nice if the lab assistant also understood the course content, so if the student gets stuck the lab assistant can help.

Who decides the scheduling of the computer labs? Would an expensive lab sit idle except for two weeks each semester, or would the computers receive daily productive use? If you have a package you would like your students to use for the six and seventh weeks of the semester but the lab is in use daily for computer classes, you are not going to be able to schedule those two weeks of intensive use in the middle of the semester unless you have a very cooperative computer department.

I think we have already addressed our next point - "Does a given particular program enhance your teaching or your student's understandings sufficiently to justify using it." Once you have decided that a program is educationally worthwhile to use, can you still justify the cost of using that software package?

Centralize campus purchase and equipment usage decisions to allow for maximum equipment use and minimum expenditures.

The last point is one that we are wrestling with at Jefferson College. Maybe a school that only has one lab with 20 computers doesn't have this problem, but we sure do. "Centralized campus purchase and equipment scheduling decisions to allow for maximum equipment usage and minimum expenditures." This goes back to the question, who decides how to schedule these labs? I have mentioned several times that we have Apple labs in active daily use, that are receiving usage because we just don't have the funds to replace them with IBM. We want to replace them and are planning to in our long-range planning, but I suspect that when it comes time that we can buy those two new IBM labs, we may still find someplace to place the Apples to allow students to continue to use them. Perhaps at that time they would go to the Arts & Sciences division for use in the English Composition classrooms for the word processing, or perhaps not. By centralizing equipment purchasing and scheduling, you get the maximum usage. This takes cooperation. Up until this point, a lot of our computer purchases have come out of department budgets; the departments, rightly so, say, "These machines are mine, and you may not use them." We are looking to pull back all or some of that equipment funding into one central pool, which would give us more capacity for buying machines. It is going to

take some sharing of machines among departments. Large volume purchase discounts can make more machines available for the same amount of money.

It is easier and more cost effective to maintain standard equipment - the gentleman asked earlier about down time. If you have three machines on campus by XYZ Corporation, it is not cost-effective to keep parts for those computers on site, and keep a technician trained in how to use them. So if they go down, they go off-site for two or three weeks to be repaired. On the other hand, if you have 200 XYZ machines, you can afford to have the parts on hand and a specially trained technician. That way you can have them up and running in 12 to 24 hours. This is a more effective and efficient way of doing things. If some machines are out for repair you can swap labs. We had a computer virus hit at the beginning of this semester; until we could get the software in to clean the virus off the machines, we had to teach some of our classes in another lab. That lab was available and we were able to use it, but we also infected that lab with the virus which was on our data disks. The labs were out of commission for about one day while we went around and cleaned the computers.

With centralized planning you can have more than one type of machine, since there is no one machine that is perfect for every instance. There is a third type of computer - the Unix-based machine - Unix is an operating system. It is not compatible with the IBM operating system (MS-DOS). There are many businesses that are using Unix-based systems. Fortunately, a personal computer with the right microprocessor can run Unix as well as MS-DOS. Your only expense is to purchase two operating systems for the one machine. There is no one perfect machine, so by centralizing your needs and by getting maximum

usage out of your labs, you do have some money freed up to buy one lab of Macintoshs to go with all of your IBM PCs, or to go out and equip one lab with the Unix operating system instead of MS-DOS. That gives you more possibilities and capabilities than if each department goes out and buys their own equipment.

Question: What should everybody have for an Associate of Arts degree? What is done at Jefferson?

Answer: Right at the moment, nothing. We strongly suggest that students take Introduction to Data Processing, but it is not a degree requirement. In the introduction class, we go over the major software, we have a unit where we talk about software in general and specific packages, and their hands-on experiences is in Appleworks, again because we have not been able to replace the Apple labs yet. When the lab goes to IBM, we will probably be using a program like PFS-First Choice. At the moment, for our transfer students and overall college requirement, there is no computer literacy requirement.

Faculty, in order to facilitate knowing what the four year schools want and expect, you need to develop networks with the faculty at the four year institution and at the other community colleges here in the state (appendix 5). If there is any particular program that your students happen to transfer to frequently, find out their specific computer requirements. The easiest way to find out what software they are using at the four year schools is to ask. You ask by knowing who the faculty members are and keeping in touch with them. One way you can help keep in touch is to read the major journals in your field, as well as general journals in the field of higher education such as the *Chronicle of Higher Education*, and the various AACJC (American Association of Community and Junior Colleges) publications. Taking note of what software and hardware reviews and advertisements appear in these journals will give you a good understanding and idea of what's being used out there.

For your career and vocational programs, your advisory boards are going to come in and tell you what the people need in your job market area to get a job, what software and

technical background. That means that you as faculty members have to stay current in the technical and non-technical aspects of your field. That includes, if a new software package comes out, you need to get your hands on it and learn it yourself, even if you don't teach it to your students. If it is the up-and-coming one package that everybody in your field is using, you need to have some hands-on experience with it.

I haven't talked a lot about continuing education or the non-degree programs, adult continuing education. The courses that are offered in computer technology are going to be dictated by what the students want to take. If you offer an Appleworks class and one person signs up for it, you are not going to offer it again. If you offer Lotus 1-2-3, it fills on the second day of registration, you scramble to offer another section, and it fills, then you are going to know that that is a popular class. The other thing with the continuing education classes is that they usually come last when scheduling class lab time. As I mentioned earlier, at Jefferson College at our North campus, adult education gets Friday nights and Sunday afternoons when credit classes are not being offered.

This concludes my presentation, and at this time I will try to answer questions.

Question: Do you have a structure in place to charge students to use the computer lab when they are not enrolled in computer classes, when they are just using word processing to type papers or things like that?

Answer: At the moment we do not. That is one area we may want to or need to look into. At the moment, our labs are available when they are open, even to non-students, to community members, and we do or did have in the past a number of people from the outside community to use our computers. We do have a few machines in our library, and those, of course, are open to the students as often as the library is open. They are not the latest or state of the art, but they are available for basic word processing. In the academic computer labs, we tell students that as long as there is not a class scheduled in there, if there is a machine available, it is theirs. That may be a source of some funding in the future.

Question: Is there a supervisor in the lab?

Answer: Yes, if the lab is open, then there is a lab assistant watching the machines. In the library, if the library is open the computers are available; you check out the software from the reserve desk, but there may or may not somebody there who knows anything about computers, in fact, there usually is not. But that is just a few machines at this point in time.

Question: Is the software already there?
Answer: In our library, no; users check out the software from the reserve desk; then they are on their own as far as using the machines. In the computer labs themselves, the lab assistant checks out to them the Appleworks package, when they work on the Apples. On the IBMs, the software is already loaded on the hard drive, and the lab assistant is available to answer questions and to hopefully make sure the users are not taking too many copies off the hard drives, putting them on the floppies, and walking out with it.

Question: What qualifications do your lab assistants have.
Answer: Most of them have been through the Introduction to Data Processing class. Most of them are computer information majors. They do not necessarily have experience on every software package. The lab supervisor, who supervises the assistants and runs the lab throughout the week, does try to have some training sessions with the lab assistants and does try to hire students that she feels will be reliable, knowledgeable and helpful to the students; but we do have a problem that our lab assistants could use better training, and they could stand to be paid more - they have to be work-study students at the moment.

Question: Are we looking to establish a particular computer literacy curriculum change?
Answer: At the moment, no. It would be nice, but right at the moment I am not aware of it. Lynn, is your committee doing anything with it? (answer unintelligible on tape)

Question: How much computer application should be part of the computer literacy class and how much computer literacy should be part of the applications class?
Answer: In an ideal world, where students could take classes for free and therefore did not care how many hours they racked up for the associates degree, they would take both. We have a course proposal which is working its way through the curriculum committees at the moment, suggesting the creation of yet a third computer literacy class for the transfer students. I have not seen the proposal so I am going on innuendo here, but this class would keep them from having to take both our Introduction to Data Processing class and our Micro-Software class, so the overall question was what should the course content be. That's something that your centralized planning committee at best, your computer information faculty at minimum, and nobody at worst, should sit down and decide. It is going to vary from school to school. You should.. find out what is going to be required of your students when they transfer. For this new course proposal, for example, one of the first questions asked was if a student who takes this class transfers to Columbia or to UMSL or to Rolla[10], will they get credit for taking it? Will it count towards their degree? From what I have heard of this new course, I'm not really sure that it will

transfer, whereas our introductory class will carry credit towards the bachelor's degree at most college and universities.

[There is a question and discussion at this point, which is unintelligible on the tape]

Let me summarize for the tape. Mineral Area College has included a computer literacy requirement in the curriculum, which merely states that the student must demonstrate that they are computer literate prior to graduation. For the transfer student that includes either credit-by-examination or actually taking their introductory computer class. On the vocational side, the definition of computer literacy is curriculum driven, and is taught usually in the introduction classes. I really would like to see what you define as computer literacy. The question is, what do you define as computer literacy if some student walks in and says, "I are lit-er-ate, I don't wanna take this class." How do you prove or disprove that.

Moberly[11] has a two hour class, for those who do not want to dive into Introduction to Computer Science or Introduction to Data Processing, that includes a terminology and basic literacy lecture component, and a lab component using Lotus, dBase, and WordPerfect. So it sounds like what you have is what most of the schools are doing. What is called the introductory computer class includes getting into a bit of the history of the computers, how the different parts work, how a computer does what it does, and why it does it. The introductory course also includes some sort of lab component. To our lab component we added an exercise where our students actually do "write" a very simple B.A.S.I.C. program, because several four-year schools wouldn't take our course in transfer credit without it. They said they wanted not only to have the applications hands-on but also a little bit of programming hands-on.

There is the question that Lynn brought up, that is currently rolling through the Arts and Sciences division at Jefferson College. How do you cover just what your students need to know without having them take this extra three hour class that does not fulfill a transfer degree requirement; it fulfills an elective, but not an actual requirement. How do you determine what is computer literacy?

I hope today that I've given you something to think about as far as computer literacy, but the definition of computer literacy is going to be driven by society expectations, be that in the transfer area by the four year schools or by the market place for the vocational students. The implication of that is that you as faculty are going to have to sit down and decide what you want your students to be able to do. The gentleman from Moberly mentioned that they want the students to complete their computer awareness class so that the English Composition teacher isn't taking two weeks out of the semester to teach a software program package. That is as it should be. On the other hand, if the math department is

going to use a particular package that nobody else on campus is, to draw and graph functions, then they should be the ones teaching that package.

All these decisions have to be made on the local basis, so I want all of you to go back to your campuses and form campus wide computer committees to determine how to best serve our students and ourselves heading into the 21st century. Thank you for coming and for your attention.

NOTES

1. These Jefferson College course numbers are: BUS 125, Introduction to Data Processing; BUS 124, BASIC Programming; CSC 137, Lotus 1-2-3 Spreadsheet Techniques; and CSC 133, Microcomputer Software Applications.

2. Classified staff are those support people who are usually paid an hourly wage: clerical and secretarial, custodial and maintenance, book store, cafeteria, etc.

3. Seymour Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, as quoted in *The Cult of Information* by Theodore Roszak, pp. 82-83. (*Mindstorms*, New York: Basic Books, 1980; *The Cult of Information*, New York: Pantheon Books, 1986)

4. "Don't trust any computer you can't lift." Steve Jobs

> In 1975 Jobs, then 19 and a technician at Atari, and Steve Wozniak, an electronics engineer at Hewlett-Packard... bought a microprocessor chip for $25 and build the first Apple... Jobs sold his Volkswagen and Wozniak pawned his calculator -- Apple Computer was born with $1300 in capital. (Deitel, Harvey and Barbara, *Computers and Data Processing*, Academic Press, Inc., 1985, pp. 334-335.)

5. Commodore's *Amiga* and an Atari machine compete against the Apple Macintosh today, but not on the level or volume of business that either company enjoyed in the initial home computer scramble.

6. Author's personal opinion, based upon reviews, articles and discussions read.

7. This quote may be apocryphal; it is not mentioned in *Father, Son and Co.*, by Thomas J. Watson, Jr.

8. Sally Greenwood Larsen, quoted in Deitel and Deitel, p. 353.

9. Three Rivers Community College, Poplar Bluff, Missouri.

10. The University of Missouri, Columbia; the University of Missouri, St. Louis; and the University of Missouri, Rolla, respectively.

11. Moberly Junior College, Moberly, Missouri

Appendix 1

BRIEF HISTORY OF THE MICROCOMPUTER

1965	Dr. John Kemeny of Dartmouth led the development of the BASIC programming language as an instructional tool.
1971 ('69?)	Invention of the microprocessor by Dr. Ted Hoff of Intel Corporation (currently disputed)
1972	The MITS Altair computer was the first commercially successful microcomputer. It sold in kit form for less than $500.
1975	Steve Jobs, 19, and Steve Wozniak build their first computer in Jobs's parent's garage in Palo Alto, California.
1977	Tandy Corporation (Radio Shack) marketed its first TRS-80. Commodore entered the market with its PET computer (named after the fad at the time -- pet rocks). Atari, the world's leader in video games, also introduced a home computer. The Apple II series computers reach the consumer.
1979	Dan Bricklin and Bob Frankston, Harvard MBA students, introduce the VisiCalc spreadsheet program, which ran on an Apple II computer. Many consider VisiCalc to be the single most important reason why microcomputers gained acceptance in the business world.
1980	Bill Gates, founder of Microsoft Corp., develops MS-DOS for the new IBM PC.
Aug. 1981	IBM introduces the IBM Personal Computer featuring the Intel 8088 16-bit micro- processor and PC-DOS (Gates' MS-DOS).
1983	Lotus 1-2-3, developed by Mitch Kapor, was introduced.
Jan. 1984	Apple introduced the Macintosh (the Lisa was introduced in late 1983).
1987	Microcomputers using the new Intel 80386 microprocessor reach the marketplace.
1988	IBM introduces PS/2 series microcomputers using OS/2 operating system.

Appendix 2

SOFTWARE EVALUATION SOURCES

The Academic Software Library handles production and distribution of software for five scholarly societies -- the American Institute of Physics, the American Political Science Association, the American Society for Engineering Education, the Center for Applied Linguistics, and the Modern Language Association. Software ranges in price from $30 to $70 and runs on MS/DOS computers.

The Academic Software Library
Box 8202, North Carolina State University
Raleigh, NC 27695-8202; (919) 737-2534
BITNET: RISLEY@NCSUPHYS

The Clearinghouse for Academic Software offers some 150 programs for DEC computers. The clearing-house is run by Digital Equipment Corporation and Iowa State University. Prices range from $100 to $1,500 for site licenses.

Clearinghouse for Academic Software
Computation Center, 297 Durham Center
Iowa State University
Ames, IA 50011; (515) 294-0323
BITNET: GMBLM@ISUMVS

CONDUIT is one of the oldest software-dissemination efforts. Although CONDUIT still sells 120 liberal-arts software packages, recent efforts have focused on four areas: English, foreign languages, mathematics and psychology. A variety of computers and operating systems are supported. Packages range in price from $50 to $500.

CONDUIT
University of Iowa, Oakdale Campus
Iowa City, IA 52242; (319) 335-4100

Cosmic is the National Aeronautics and Space Administration software-dissemination venture, run by the University of Georgia. Most of its 1,200 software packages were developed by or for NASA's research teams. The median price for microcomputer software is $250, the median price for mainframe software is $1,000, and the discount for higher education is at least 50%.

Cosmic, University of Georgia
382 East Broad Street
Athens, GA 30602; (404) 542-3265
BITNET: COSMIC@UGA

ISAAC - Information System for Advanced Academic Computing - is a free electronic data base that contains information about 3,400 higher-education software packages for MS/DOS computers. The data base includes the name and purpose of the software, the hardware setup it needs, the cost, and where to get it. Emphasis is placed on peer-review information to help evaluate software packages.

ISAAC Access, MS FC06
University of Washington
Seattle, WA 98195; (206) 543-5604
BITNET: ISAAC@UWAEE

National Collegiate Software is a division of Duke University Press. Most of the software is for the social sciences, but some English and journalism software is also available. The press provides peer review and quality checks.

National Collegiate Software
Duke University Press
6697 College Station
Durham, NC 27708; (919) 684-2173
order line (919) 684-6837

Project BioQUEST is a software developers' consortium, a group of people who write biology software for instruction. The 18 BioQUEST software modules run on Apple Macintosh computers.

BioQUEST, Department of Biology
Beloit College
700 College Street
Beloit, WI 53511; (503) 345-7749

Project SERAPHIM has been collecting and offering chemistry software -- primarily for instruction in high school and college -- since it was set up with a grant from the Department of Education in 1982. The 200 disks in the SERAPHIM catalogue contain more than 600 chemistry programs for Apple II, Apple Macintosh, Commodore, and MS/DOS computers. Disks are $5 each, with postage and handling costing another $5 a disk.

Project SERAPHIM, Department of Chemistry
University of Wisconsin
Madison, WI 53706; (608) 263-2837

PSRC Software offers three packages developed at the Population and Society Research Center at Bowling Green University. The center hopes to come out with more software this year, focusing on instructional packages.

PSRC Software, 302 Hayes Hall
Bowling Green University
Bowling Green, OH 34303; (419) 372-8648

Wisc-Ware is a distribution network for research and instructional software developed for MS/DOS computers. Colleges and universities that join the Wisc-Ware consortium get lower prices on the software and some technical assistance. They also have the right to submit software for Wisc-Ware to distribute. Wisc-Ware currently offers 287 programs in many disciplines.

Wisc-Ware, Academic Computing Center
University of Wisconsin at Madison
1210 West Dayton Street
Madison, WI 53706; (800) 543-3201
BITNET: WISCWARE@WISCMACC

The above information is extracted from the article "Colleges, Scholarly Societies, and Foundations Create Software-Dissemination Projects to Share Expertise," *The Chronicle of Higher Education*, October 17, 1990, pp. A17, A19 - A20. There are other software-dissemination services not included in this limited survey.

Appendix 3

Guide to Word Processors

* Simple word processors:

  - should be able to move, copy, delete and insert text,
  - should include a spelling check routine,
  - should be able to print underline, italic, boldface, and variable characters per inch.

* Full-featured word processors should additionally include:

  - automatic foot- and end-note placement
  - automatic index-creation
  - basic graphics capabilities (boxes, lines, shading)
  - access to laser printer for final document

* Desktop publishing software and equipment should include:

  - layout templates
  - scanner and full-feature graphics
  - high-resolution monitor for on-screen layout and editing

Appendix 4

Software Acquisition Check List

* Learn which software packages are standard in your field.

  - Would using this software enhance, or distract from, your course?

  - Does the learning experience provided by the software outweigh the time needed to learn to use the software effectively?

  - On which machine(s) does this software work?

* How many other instructors would also need access to this type of computer during your class time?

  - Would students have adequate access and knowledgeable support to use this package outside of class time?

  - Who decides the scheduling of the computer laboratories? Would an expensive laboratory sit idle except for two weeks each semester, or would the computers have daily productive use?

Appendix 5

Computer Needs by Division

* Transfer degree programs:

  - The need to learn computer technology is dictated by the expectations of the baccalaureate programs to which your students most frequently transfer.

  - Faculty need to develop collegial networks with faculty both at four-year institutions and at other community colleges.

  - Faculty should read software and hardware reviews and note advertisements in the major journals in their fields.

* Non-transfer degree programs:

  - The need to learn computer technology is dictated by the expectations of the job market.

  - Faculty need to stay current in both technical and non-technical aspects of their field.

* Continuing education non-degree programs:

  - The need to learn computer technology is dictated by the expectations of your prospective students.

  - Access to computer laboratories will also dictate what courses can be offered.

Appendix 6: additional lecture notes included in handout given to attenders

Check List for Computer Use

* Know what the computer requirements of four-year colleges and/or industry are for your graduates.

How important it is to include computers in the classroom depends upon the department, the subject being taught, the level of the class, and the demands of the job market. Computer usage and literacy levels in the Vocational-Technical curriculums will be dictated by the job market -- the "real" world. Therefore, there are few problems (primarily funding) in deciding appropriate technology to use in non-transfer degree programs.

In the transfer degree programs, appropriate usage will vary widely from department to department and between courses within departments. Faculty need to know what the four-year schools require in computer literacy -- what level of knowledge they expect their juniors to have already in major coursework, and what skills the senior institutions will teach in the upper-division classrooms. Faculty should also keep up-to-date in the "real world" computer usage required for jobs their students will enter after earning a four-year degree. Like it or not, we are all vocational-oriented, whether preparing our students for the work force following two years of college or following advanced graduate degrees.

* Encourage the use of word processing software in producing written projects.

The typewriter has been replaced by the word processor in today's business and academic worlds. Whether our curriculum is vocational or academic, everyone needs to be able to use word processing technology. **This includes <u>you</u>**! Community college faculty are mostly sheltered from the intense "publish or perish" academic world, but we still generate large amounts of paper each semester.

Encourage, nay, <u>insist</u> upon word-processed major papers, if there is adequate access to computers for your students. You should learn the word processing package that your students will use, at least to the level you will expect them to use it. Such a package should be easy to learn and relatively unsophisticated -- it is replacing the typewriter, not the publishing industry. Include early rough drafts of major papers in the grading criteria, critique these drafts with the student, and insist upon polished final copies **with no "cut and paste" or white-out** on the pages!!!

* Be realistic about student access to computer equipment.

In an ideal world, every student, faculty and staff member would have their own state-of-the-art computer and printer. In reality, current demand for computer classes dictate that computer laboratories be scheduled as classrooms, to be used during all hours of the day and night (including Saturdays and Sundays), leaving little or no access to machines outside class. Computer literacy ideals must be moderated by machine accessibility.

* Does any one program enhance your teaching or your students' understanding sufficiently to justify using it?

Faculty must be realistic about the benefits of using specialized instructional software packages. You may have found a program that perfectly illustrates a difficult concept; but if it takes only 15 minutes to work through the program (out of the entire semester of instruction), perhaps an in-class demonstration on large screen will be a better use of time and resources than purchasing 25 copies of the program and having 200 students try to get laboratory time in the course of one week during the semester.

* Centralize campus purchase and equipment scheduling decisions to allow for maximum equipment use and minimum expenditures.

Funding of adequate computer facilities remains the greatest obstacle to teaching all how need, and want, to learn these new skills. Two campus sources need to be involved in this centralized decision-making: a central computer administrator and her/his staff, and a campus-wide committee of faculty and staff from all areas to advise that administrator of their needs and to be computer information resources for their respective constituents.

Too often, computers are ordered and a laboratory is set up; but no thought is given to long-range computer usage -- what software will be needed or used, or who will staff the laboratories and support maintenance of the equipment. The expensive equipment sits unused, or, worse, is taken over by student computer-jocks. As long-range planning is vital in every other area in the college, so, too, planning is needed to provide adequate and appropriate computer technology to all our students and our community members.

Benefits from central planning of computer technology includes:

- large-volume purchase discounts can provide more machines for the funds available;
- it is easier and more cost efficient to maintain standard equipment;
- laboratories can be "swapped" when some machines are out for repairs;
- all institutional needs can be met, not by one "perfect" machine but by justifying the need for two or more different systems through planning for their near-constant and appropriate use.

ERIC Clearinghouse for
Junior Colleges

JUN 21 1991